NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

EDGE COMPLETION SEQUENCES FOR CLASSES OF CHORDAL GRAPHS

by

Richard M. Odom

June 1995

Thesis Advisor: Craig W. Rasmussen

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE June, 1995

3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE EDGE COMPLETION SEQUENCES FOR CLASSES OF CHORDAL GRAPHS

5. FUNDING NUMBERS

6. AUTHOR(S) Odom, Richard M.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Given an incomplete graph $G = (V, E)$ of order n and size m and possessing some property P, a *P-Completion sequence* for G is a sequence $e_1, \ldots, e_s$ of edges, where $s = \binom{n}{2} - m$, with the property that if $G_0 = G$ then (1) G_i is obtained from G_{i-1} by insertion of exactly one edge and (2) G_i has property P for each $1 \leq i \leq s$. The $O(n^2)$ algorithm for constructing chordal completion sequences depends on a perfect elimination ordering α. We show that completion sequences can be generated for several subclasses of chordal graphs using a generic algorithm, of which the chordal completion algorithm is a special case, with the input being a graph from one of the subclasses and a specific ordering of the vertices that characterizes the subclass. We include a discussion of the strong elimination ordering for strongly chordal graphs, the interval elimination ordering for interval graphs and the bicompatible ordering for unit interval graphs. We define a threshold elimination ordering and then prove that a graph G is a threshold graph if and only if G has a threshold elimination ordering. We show that completion sequences exist for strongly chordal graphs. Finally, we prove that completion sequences for strongly chordal, interval, unit interval, split and threshold graphs can be contructed in $O(n^2)$ time.

14. SUBJECT TERMS Chordal Graphs, Edge Completion Sequences, Eliminaition Orderings

15. NUMBER OF PAGES 52

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFI CATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for public release; distribution is unlimited

EDGE COMPLETION SEQUENCES FOR CLASSES OF CHORDAL GRAPHS

Richard M. Odom
Lieutenant, United States Navy
B.S., San Diego State University, 1988

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
June, 1995

Author: Richard M Odom LT, USN

Richard M. Odom

Approved by: Craig W. R

Craig W. Rasmussen, Thesis Advisor

David E. Mann, Second Reader

Richard Franke

Richard Franke, Chairman, Department of Mathematics

ABSTRACT

Given an incomplete graph $G = (V, E)$ of order n and size m and possessing some property P, a *P-completion sequence* for G is a sequence $e_1, \ldots, e_s$ of edges, where $s = \binom{n}{2} - m$, with the property that if $G_0 = G$ then (1) G_i is obtained from G_{i-1} by insertion of exactly one edge and (2) each G_i has property P for each $1 \leq i \leq s$. The $\mathcal{O}(n^2)$ algorithm for constructing chordal completion sequences depends on a perfect elimination ordering α. We show that completion sequences can be generated for several subclasses of chordal graphs using a generic algorithm, of which the chordal completion algorithm is a special case, with the input being a graph from one of the subclasses and a specific ordering of the vertices that characterizes the subclass. We include a discussion of the strong elimination ordering for strongly chordal graphs, the interval elimination ordering for interval graphs and the bicompatible ordering for unit interval graphs. We define a threshold elimination ordering and then prove that a graph G is a threshold graph if and only if G has a threshold elimination ordering. We show that completion sequences exist for strongly chordal graphs. Finally, we prove that completion sequences for strongly chordal, interval, unit interval, split and threshold graphs can be constructed in $\mathcal{O}(n^2)$ time.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

A. DEFINITIONS

A *graph* is a pair $G = (V, E)$, where V is a finite set of elements called *vertices* and E is a set of unordered pairs of vertices called *edges.* A *digraph* or *directed graph* $D = (V, A)$ is similarly defined, but in this case A is a set of *ordered* pairs of vertices. We will assume that all graphs are undirected and without loops or multiple edges (see Figure 1).

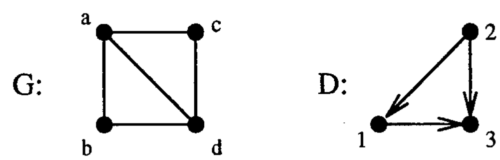

Figure 1. A graph G and a digraph D.

The *order* of G is the number of vertices, denoted $|V|$, and the *size* of G is the number of edges in G, denoted $|E|$. In this paper the set notation will not be used when referring to an edge. Therefore, $xy \in E$ and $\{x, y\} \in E$ will have the same meaning. A *path* is a sequence of vertices $v_1, v_2, \ldots, v_n$ with the property that $v_i v_{i+1} \in E$ for each i, $1 \leq i \leq n-1$. The *length* of a path from from v_1 to v_n in G is the number of edges on the path. A path is *simple* if no vertex occurs more than once in the sequence $v_1, v_2, \ldots, v_n$. If $v_1, v_2, \ldots, v_n$ is a simple path, then we say that $v_1, v_2, \ldots, v_n, v_1$ is a *cycle.* A graph $G = (V, E)$ is *connected* if there is a path between each pair of vertices of G. A graph G is called a *forest* if it contains no cycles, and a connected forest is a *tree* (see Figure 2).

A graph $H = (W, F)$ is a *subgraph* of a graph $G = (V, E)$, denoted $H \leq G$, if $W \subseteq V$ and $F \subseteq E$. If $H \leq G$, then G is a *supergraph* of H. An *induced subgraph* of $G = (V, E)$ is a graph $H' = (W', F')$ where W' is a nonempty subset of V and the set F' consists of those edges of G incident with two vertices of W' (see Figure 3). If $A \subseteq V$ we denote by G_A the subgraph of G induced by the set A. The *complement*

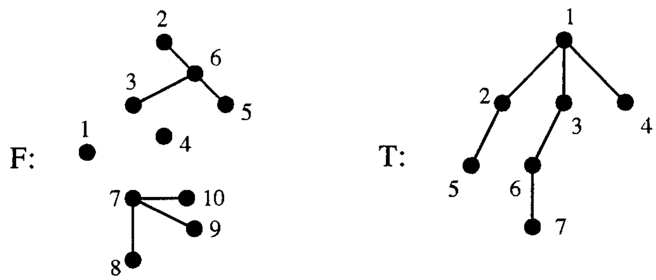

Figure 2. A forest F and a tree T.

of a graph $G = (V, E)$ is the graph $G^c = (V, E^c)$ where $xy \in E^c$ if and only if $xy \notin E$. See Figure 4 for an illustration.

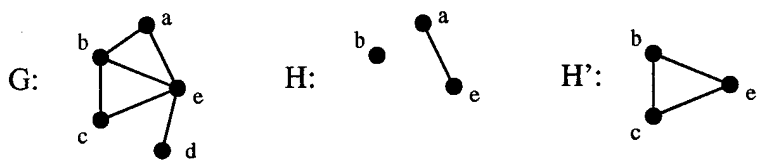

Figure 3. A graph G, subgraph H, and induced subgraph H'.

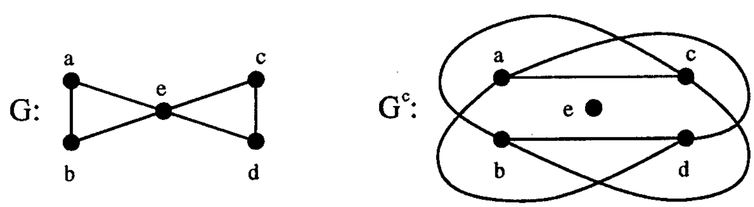

Figure 4. A graph G and its complement G^c.

If $e = xy$ is an edge of G, then x is said to be *adjacent* to y, y is adjacent to x, x and y are *incident* to e and e is incident to x and y. The *degree* of a vertex v, denoted $\deg(v)$, is the number of edges incident to v. A *complete graph* on n vertices, denoted K_n, is a graph in which every pair of distinct vertices is adjacent. Let $G = (V, E)$ be a graph, and let $v \notin V$ be a vertex. The *join*, $G + \{v\}$, is the graph obtained by adding all edges between the vertices of V and v.

The set of all vertices adjacent to x is called the *open neighborhood* of x and is denoted $N(x)$. The *closed neighborhood* of x is $N[x] = N(x) \bigcup \{x\}$. A *clique* in a graph G is a subgraph of G that is complete. A *maximal clique* in G is a clique in

G that is contained in no larger clique. The *clique number*, $\omega(G)$, is the order of a largest clique. A set of vertices in G form an *independent set* in G if there are no edges between any of the vertices in the set. The *independence number*, $\alpha(G)$, is the order of a largest independent set of G. An *n-coloring* of a graph G is an assignment of n colors (colors are elements of some set) to the vertices of G in such a way that adjacent vertices are assigned different colors. The minimum number of colors needed to color a graph G is called the *chromatic number* of G, and is denoted by $\chi(G)$ (see Figure 5). A *clique cover* of size k is a partition of the vertices as $V = A_1 \bigcup A_2 \bigcup \ldots \bigcup A_p$ such that A_i is a clique for each $1 \leq i \leq p$. The clique cover number, $k(G)$, is defined to be the smallest number of cliques that contain all vertices of G. A vertex $v \in V$ is *simplicial* in G if $N(v)$ is a clique, i.e., the neighbors of v form a complete subgraph. A graph G is *chordal* if it contains no cycles of order greater than three.

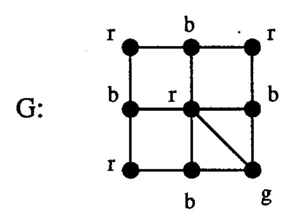

Figure 5. A 3-coloring of G; $\chi(G) = 3$.

B. PERFECT GRAPHS

All of the classes of graphs discussed in this paper are *perfect graphs.* A graph $G = (V, E)$ is perfect if it satisfies the following properties:

$$\omega(G_A) = \chi(G_A) \text{ for all } A \subseteq V \tag{I.1}$$

$$\alpha(G_A) = k(G_A) \text{ for all } A \subseteq V \tag{I.2}$$

$$\omega(G_A)\alpha(G_A) \geq |A| \text{ for all } A \subseteq V \tag{I.3}$$

Perfect graphs were introduced by Berge [Ref. 1] in the early 1960s. He conjectured that properties I.1 and I.2 were equivalent. The equivalence of these two properties became known as *The Perfect Graph Conjecture.* In 1972 Lovász [Ref. 2] showed that not only were the first two properties equivalent, but that I.3 was equivalent to the first two. The equivalence of these properties became known as *The Perfect Graph Theorem.* Much work has been done to show that many classes of graphs are perfect. Chordal graphs were one of the first classes of graphs to be proven to be perfect. In the following pages we introduce classes of chordal graphs that will be useful in subsequent chapters. See Figure 7 for examples of perfect graphs. All of them are chordal except the permutation graph. For a detailed analysis of perfect graphs, see Golumbic [Ref. 3].

Given a family $\mathcal{F} = \{F_1, F_2, \ldots, F_k\}$ of subsets of some universal set $\mathcal{U}$, the *intersection graph* of $\mathcal{F}$ is the graph $G = (\mathcal{F}, E)$ in which the vertex set V corresponds to the elements of $\mathcal{F}$ and there is an edge $F_iF_j \in E$ if and only if $F_i \cap F_j \neq \emptyset$. Marczewski [Ref. 4] showed that every graph is the intersection graph of some family of sets, so this by itself is not such an interesting property. Of interest are graphs that are the intersection graphs of particular families of sets. An *interval graph* is a graph $G = (V, E)$ that can be represented as the intersection graph of a set of intervals on the real line. If the intervals are all the same length, then G is said to be *unit interval.* A *circular-arc graph* is a graph that can be represented as the intersection graph of arcs on a circle. Fulkerson and Gross [Ref. 5] give a characterization of interval graphs in terms of the maximal cliques in a graph G. They show that G is an interval graph if and only if the maximal cliques of G can be ordered in such a way that, for any vertex $v \in V$, the maximal cliques containing v occur consecutively.

A graph $G = (V, E)$ is a *split graph* if there is a partition $V = K \bigcup I$ of the vertex set such that K is a clique and I is an independent set. In general the partition of V will not be unique. Another characterization of split graphs is given in terms of *degree sequences.* Let $G = (V, E)$ be a graph, and let $\mathcal{D} = (v_1, \ldots, v_n)$ be an ordering

of the vertex set V such that for $i > j$, $\deg(v_i) \leq \deg(v_j)$. The ordering $\mathcal{D}$ is called the degree sequence of G. Hammer and Simeone [Ref. 6] state the next theorem that allows us to determine if a graph G is a split graph based on its degree sequence.

THEOREM I.1 (Hammer, Simeone) *Let $G = (V,E)$ be a graph with degree sequence $\mathcal{D} = (v_1, \ldots, v_n)$, and let $m = max\{i | deg(v_i) \geq i - 1\}$. Then G is a split graph if and only if*

$$\sum_{i=1}^{m} deg(v_i) = m(m-1) + \sum_{i=m+1}^{n} deg(v_i).$$

Furthermore, if this is the case then $\omega(G) = m$.

If $G = (V, E)$ and $e = xy \in E$, an *orientation* of e is an assignment of *direction*, i.e., e is replaced by either (x, y) or (y, x). An orientation of G is an orientation of all edges of G. A graph $G = (V, E)$ is a *comparability graph* if there exists a *transitive orientation* of its edges, i.e., $\mathcal{T}$ is a transitive orientation for G if for any vertices $x, y, z \in V$, if $(x, y), (y, z) \in \mathcal{T}$ then $(x, z) \in \mathcal{T}$. A subclass of the comparability graphs is the class of *permutation graphs.* Given a permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ of $\{1, 2, \ldots, n\}$, denote the position of i in π, by ${\pi_i}^{-1}$. An *inversion* is a pair $\{i, j\} \in \{1, 2, \ldots, n\}$ with the property that $i < j$ but ${\pi_i}^{-1} > {\pi_j}^{-1}$. Construct the graph $G(\pi) = (V, E)$ with $V = \{1, 2, \ldots, n\}$ and $E = \{ij | \{i, j\}$ is an inversion in $\pi\}$. The graph $G(\pi)$ is called a permutation graph. A characterization of permutation graphs in terms of comparability graphs is given by Pnueli, Lempel and Even [Ref. 7]. They show that a graph G is a permutation graph if and only if G and G^c are comparability graphs.

A graph $G = (V, E)$ is a *threshold graph* if there exists an integer labeling $\mathcal{C} = \{c_1, \ldots, c_n\}$ of V and an integer (threshold) t such that for distinct vertices c_i and c_j, $c_i c_j \in E$ if and only if $c_i + c_j > t$. Chvátal and Hammer [Ref. 8] modify this definition in order to define a threshold graph in terms of a *degree partition* of its vertex set, and they show that the two definitions are equivalent. Let $G = (V, E)$ and

let $0 < \delta_1 < \delta_2 < \ldots < \delta_m < |V|$ be the degrees of the nonisolated vertices. Define $\delta_0 = 0$. The degree partition of V is given by

$$V = D_0 + D_1 + \ldots + D_m, \tag{I.4}$$

where D_i is the set of all vertices of degree δ_i. The set D_0 may be empty. A graph $G = (V, E)$ with the degree partition described in Equation I.4 is a threshold graph if for every pair of distinct vertices $x, y \in$ V, $x \in D_i$, $y \in D_j$, $xy \in E$ if and only if $i + j > m$ (see Figure 6). Golumbic [Ref. 3] discusses the relationship between threshold graphs and other classes of graphs. Since the vertices of a threshold graph can be partitioned into an independent set and a clique, then a threshold graph is also a split graph. In figure 6, $K = \{4, 9, 6, 3\}$ and $I = \{7, 2, 10, 8, 5, 1\}$.

G:

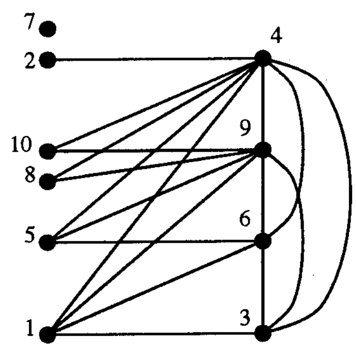

Figure 6. Threshold graphs are also split graphs.

Another characteristic of threshold graphs is that they can be transitively oriented. Let G be a threshold graph, and label the vertices of G as $1, 2, \ldots, n$ such that if $\deg(i) < \deg(j)$ then $i < j$. Orient each edge toward its larger numbered endpoint. This yields a transitive orientation of G. Therefore, threshold graphs are also comparability graphs. Threshold graphs can also be characterized in terms of permutation graphs. Let $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ and $\eta = (\eta_1, \eta_2, \ldots, \eta_m)$ be two sequences. The *shuffle product* $\sigma \sqcup\!\sqcup\, \eta$ is the set of sequences of the form $(\sigma^1\eta^1 \cdots \sigma^k\eta^k)$ where σ^i and η^i are subsequences of σ and η, respectively. Golumbic [Ref. 3] shows that G is a threshold graph of order n in which k vertices are independent and the

remaining $n-k$ vertices constitute a clique if and only if G is the permutation graph $G(\pi)$ where $\pi \in (1,2,\ldots,k) \sqcup\sqcup (n,n-1,\ldots,k+1)$.

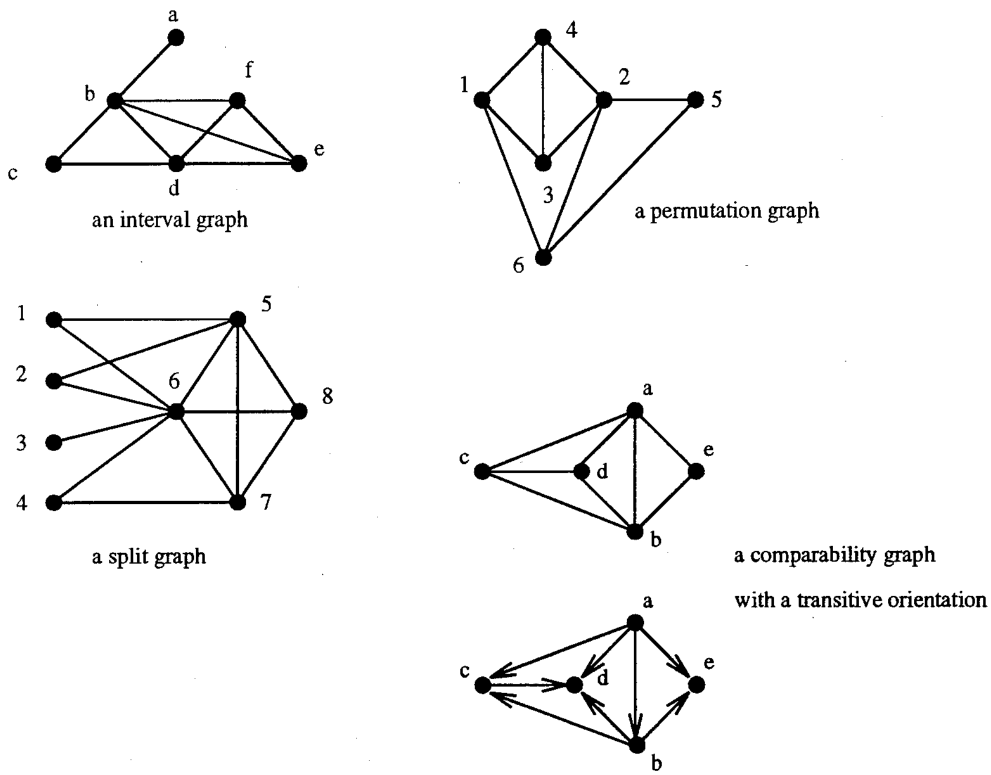

Figure 7. different types of perfect graphs.

C. COMPLEXITY OF ALGORITHMS

In Chapter III we present algorithms that produce completion sequences for different classes of chordal graphs. An algorithm is a general step-by-step procedure that will solve a given problem. Algorithms that are used most often are ones that can solve problems in a *reasonable* amount of time. Given a problem Π and an algorithm $\mathcal{A}$, we wish to determine an upper bound for the amount of time it will take a computer to solve Π using $\mathcal{A}$. With this upper bound in hand, we can then determine whether or not the amount of time needed to solve Π is reasonable or not.

Suppose we have a problem Π and an algorithm $\mathcal{A}$ that solves Π. Let the function $g(m)$ denote the number of computational steps needed to solve Π where m

is the size of the input to Π. If there exists a function $f(m)$ and constants c and k such that $|g(m)| \leq c|f(m)|$, for $m \geq k$, then we say that $g(m)$ is *dominated* by $f(m)$. Another way to express this relationship is to use *big-Oh* notation. If $g(m)$ is dominated by $f(m)$, then we write $g(m) = \mathcal{O}(f(m))$, where $\mathcal{O}(f(m))$ is read *order f* or *big-Oh of f*. Thus, the *computational complexity* of $\mathcal{A}$ is written as $\mathcal{O}(f(m))$.

A *polynomial time algorithm* is defined to be one whose computational complexity is expressed as $\mathcal{O}(p(n))$, where the complexity function p is a polynomial and n denotes the input length. When comparing the growth rates of polynomial complexity functions (e.g., n, n^2, n^3) to other types of non-polynomial complexity functions (e.g., $n!, 2^n, e^n$), we see that for large n the growth rates of the non-polynomial complexity functions are much faster than the growth rates for the polynomial complexity functions. For this reason polynomial time algorithms are more efficient because they can generally solve problems with large input in a comparatively reasonable amount of time.

We often characterize problems by the complexity of the most efficient algorithms that are known to solve them. It is generally accepted, although not proven, that *NP-complete* problems cannot be solved by an efficient algorithm (i.e., polynomial time). For example, determining the chromatic number of an arbitrary graph is a problem for which it is believed can only be solved by an inefficient non-polynomial time algorithm. We will not deal with NP-complete problems in this paper. In fact we will show that the algorithms in Chapter III are polynomial time algorithms. More on the theory of NP-completeness can be found in Garey and Johnson [Ref. 9] and Golumbic [Ref. 3].

D. GRAPH COMPLETIONS

Let $G = (V, E)$ be any graph, and let k be an integer. Let P be a property held by some graphs, and suppose G does not have property P. The *graph completion problem* can be stated as follows: Can at most k edges be added to G in order to

obtain a graph that has property P? This problem and its variations have been well-studied. Below is a brief description of four different completion problems with the focus of this paper being the conditional graph completion problem.

1. Hamiltonian Completion Problem

A *hamiltonian cycle* is a simple cycle in a graph $G = (V, E)$ that contains all vertices of G. If a graph G has a hamiltonian cycle, then G is a *hamiltonian graph.* For more on hamiltonian graphs see Chartrand and Lesniak [Ref. 10] or Roberts [Ref. 11]. The hamiltonian completion problem, studied by Goodman and Hedetniemi [Ref. 12], is an example of a graph completion problem. This type of completion problem is a special case of the famous Traveling Salesman Problem in which each edge in a graph G is assigned a weight of 0 and each edge in K_n not in G is assigned a weight of 1. Goodman and Hedetniemi [Ref. 12] define the hamiltonian completion problem as follows: given a graph $G = (V, E)$, what is the minimum number of edges that need to be added to G in order to make the resulting graph hamiltonian? In order to solve this problem Goodman et al. [Ref. 12] define a parameter called the *hamiltonian completion number*, $\mathrm{hc}(G)$, to be the minimum number of edges that need to be added to G to make the resulting graph hamiltonian. This is a well-defined parameter, since K_n is hamiltonian for all $n > 2$. Examples of other types of graph completion problems are the interval graph completions studied by Garey, Gavril, and Johnson [Ref. 13], and the path graph completions studied by Gavril [Ref. 14].

2. Sandwich Properties for Graphs

The Graph Sandwich Problem was studied by Golumbic, Kaplan, and Shamir [Ref. 15]. This problem arises in such diverse areas as sychronizing parallel processes, physical mapping of DNA, temporal reasoning, phylogenetic trees and sparse systems of linear equations. Suppose $G = (V, E)$ and $H = (V, F)$ are two graphs such that $H \leq G$. A property P is called a *sandwich property* if there exists a graph $G' = (V, E')$ having property P such that $H \leq G' \leq G$. Notice that G and H need not possess

property P. In fact a nontrivial sandwich problem requires that neither G nor H possess property P. The graph sandwich problem can be stated as follows: Given two graphs $G = (V, E)$ and $H = (V, F)$, $H \leq G$, does there exist a graph $G' = (V, E')$ having property P such that $H \leq G' \leq G$? The graph sandwich problem is a different kind of completion problem because there are certain restrictions that are placed on the graph G'. G' must include certain edges, while certain edges are forbidden. However, there is a certain amount of freedom in deciding to include any subset of the remaining edges. Split graphs and threshold graphs are examples of classes of perfect graphs that possess the sandwich property.

3. Squeeze Properties For Graphs

A completion problem similar to the graph sandwich problem was studied by Lick and White [Ref. 16]. Suppose $G = (V, E)$ and $H = (V, F)$ are two graphs such that $H \leq G$. A property P is called a *squeeze property* if whenever G and H have property P and $H \leq G' \leq G$ for some graph G', then G' also has property P. The squeeze problem can be stated as follows: Given two graphs $G = (V, E)$ and $H = (V, F)$, where $H \leq G$ and G and H possess property P, does there exist a graph $G' = (V, E')$ having property P such that $H \leq G' \leq G$? The main difference between the sandwich problem and the squeeze problem is that the squeeze problem requires G and H to have property P, while the sandwich problem does not. Well-known parameters that give rise to squeeze properties are the chromatic number, $\chi(G)$, and the clique number, $\omega(G)$. Squeeze properties related to the chromatic number were studied by Lick and White [Ref. 17].

4. Conditional Graph Completion Problem

Conditional graph completion problems are yet another family of completion problems that are different than the completion problems mentioned earlier, and are the subject of the research described here. The idea of a conditional graph completion was first studied by Grone et al. [Ref. 18] in order to gain insight into conditions

under which the following completion problem can be solved: Given a partial positive definite Hermitian matrix, can it be completed to a positive definite matrix? Here a *partial matrix* is an $m \times n$ matrix where certain entries are known elements from some specified set $\mathcal{S}$, while the remaining entries are free to be any element from $\mathcal{S}$. A *conventional matrix*, on the other hand, is defined in Johnson [Ref. 19] to be a matrix in which all entries are specified. A partial Hermitian matrix is an $n \times n$ partial matrix in which the specified entries are from the set of complex numbers. If the (i, j) entry is known, then the (j, i) entry is also known and the two entries are complex conjugates of each other, i.e., if $(i, j) = a + bi$, then $(j, i) = a - bi$. Let $\mathcal{A} = (a_{ij})$ be an $n \times n$ partial Hermitian matrix. A *completion* of $\mathcal{A}$ is an $n \times n$ conventional matrix $\mathcal{B} = (b_{ij})$ such that if a_{ij} is specified then $b_{ij} = a_{ij}$. Let $\mathcal{C}$ be an $n \times n$ symmetric matrix. Then $\mathcal{C}$ is a *positive definite matrix* if and only if $x^t\mathcal{C}x > 0$ for all $x \in \Re^n$, $x \neq 0$. Another way to express the problem studied by Grone et al. [Ref. 18] is as follows: Given an $n \times n$ partial positive definite Hermitian matrix $\mathcal{A} = (a_{ij})$ can a specification of the unspecified entries be found so that the resulting matrix $\mathcal{B} = (b_{ij})$ is positive definite? This type of problem is known as a *matrix completion problem.* A general matrix completion problem asks whether a partial matrix has a completion with a certain property of interest. For a survey of matrix completion problems see Johnson [Ref. 19].

In a sense, the conditional graph completion problem is the graph theoretic analog to the matrix completion problem. The conditional graph completion problem is slightly more restrictive in that each successive supergraph is required to have the property being questioned. The conditional graph completion problem can be stated as follows: Given a graph $G = (V, E)$, of order n and size m, with property P, is there a sequence of edges $e_1, e_2, \ldots, e_k$, where $k = \binom{n}{2} - m$, that can be added to G in such a way that each successive supergraph has property P? Such a graph is called *P-completable*, and the sequence of edges $e_1, \ldots, e_k$ is called a *P-completion sequence*, or simply a *completion sequence* when no ambiguity exists. We will sometimes refer to

the resulting sequence of graphs $G_1, \ldots, G_k$ as the completion sequence. If all graphs with property P are P-completable, we say the class Π of graphs with property P is a *completion class.* Grone et al. [Ref. 18] developed an efficient algorithm that determines the completion sequence for a graph G if and only if G is chordal. This chordal completion algorithm depends on a particular ordering of $V(G)$. Rasmussen [Ref. 20] showed that completion sequences exist for interval, unit interval, split, circular-arc, comparability, permutation, strongly chordal, and threshold graphs. In Chapter III we present generic algorithms, of which the chordal completion algorithm is a special case, that can be used in conjunction with specialized vertex orderings to find completion sequences for strongly chordal, interval, unit interval, split, and threshold graphs. For split graphs we use the degree sequence of the vertex set. For strongly chordal, interval, unit interval, and threshold graphs, we use vertex *elimination orderings* specific to each class. In Chapter II we discuss these specific elimination orderings and give the complexities of the algorithms used to generate them.

II. ELIMINATION ORDERINGS

A. CHORDAL GRAPHS

It has been shown by Dirac [Ref. 21] that every chordal graph has a simplicial vertex. This fact was used by Rose [Ref. 22] to show that every chordal graph has a *perfect elimination ordering.* A perfect elimination ordering is a bijection $\alpha : V \to \{1, 2, \ldots, n\}$ that assigns a label to each vertex in V in such a way that, for each $1 \leq i \leq n$, v_i is simplicial in $G - \{v_1, v_2, \ldots, v_{i-1}\}$. See Figure 8 for an illustration.

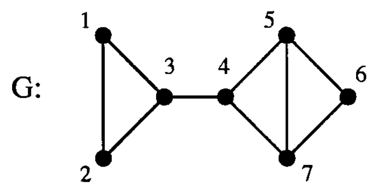

Figure 8. α = (1,2,3,4,5,6,7) is a perfect elimination ordering.

A polynomial time algorithm which uses *lexicographic breadth-first-search* for finding a perfect elimination ordering for chordal graphs is presented in Leuker, Rose, and Tarjan [Ref. 23]. If $G = (V, E)$ is a graph of order n and size m, then the complexity of the algorithm given in [Ref. 23] is $\mathcal{O}(n + m)$. Given a chordal graph G and a perfect elimination ordering α, Grone, Johnson, et al. [Ref. 18], define a *chordal completion sequence* in the following way. Let $v_k = \alpha^{-1}(k), k = 1, 2, \ldots, n$, and let s be the number of edges missing from G, i.e., if $|E| = m$, then $s = \binom{n}{2} - m$. Define the sequence of graphs $G_0, G_1, \ldots, G_s$ using Equations II.1-II.6:

$$G_0 = G \tag{II.1}$$

$$k_i = \max\{j | deg(v_j) < n - 1\} \tag{II.2}$$

$$j_i = \max\{l | v_{k_i} v_l \notin \mathrm{E}_{i-1}\} \tag{II.3}$$

$$e_i = v_{k_i} v_{j_i} \tag{II.4}$$

$$E_i = E_{i-1} \bigcup \{e_i\}, \ i = 1, 2, \ldots, s \tag{II.5}$$

$$G_i = (V, E_i) \tag{II.6}$$

Grone, Johnson, et al. [Ref. 18] showed that if α is a perfect elimination ordering for G_0, then α is a perfect elimination ordering for G_i, $i = 1, 2, \ldots, s$. The generic algorithms we present in Chapter III are essentially the same as the algorithm given by Equations II.1-II.6 except that the ordering of the vertices of the input graph G is arbitrary. The rest of this chapter is devoted to the discussion of the specific elimination orderings that correspond to strongly chordal, interval, unit interval and threshold graphs.

B. STRONGLY CHORDAL GRAPHS

Strongly chordal graphs were characterized by Farber [Ref. 24] in terms of *strong elimination orderings.* A strong elimination ordering of a graph $G = (V, E)$ is a bijection $\theta : V \to \{1, 2, \ldots, n\}$ with the property that for each i, j, k, and l, if $i < j$, $k < l$, $l \in N[i]$, and $k \in N[j]$, then $l \in N[j]$ (see Figure 9). Farber [Ref. 24] showed that there exists a strong elimination ordering for a graph G if and only if G is *strongly chordal.* If we consider the case $i = k$ in the preceding definition, then if $i < j$ and $i < l$, where $l \in N[i]$, and $i \in N[j]$, then $l \in N[j]$. This forces i to be simplicial. Thus, a strong elimination ordering is also a perfect elimination ordering, and a strongly chordal graph is also chordal.

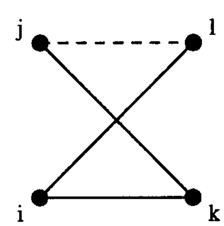

Figure 9. Given a strong elimination ordering, if $i < j$, $k < l$ then jl is forced.

Just as chordal graphs can be characterized by simplicial vertices, Farber [Ref. 24] characterized strongly chordal graphs in terms of *simple* vertices. Let $G = (V, E)$ be any graph. Two vertices u and v are *compatible* in G if $N[u] \subseteq N[v]$ or $N[v] \subseteq N[u]$.

Otherwise, u and v are called *incompatible.* A vertex v is defined to be simple in G if the vertices in $N[v]$ are pairwise compatible or, equivalently, if the set $\{N[u] | u \in N[v]\}$ can be linearly ordered by inclusion. See figure 10 for an illustration. In G vertex x is both simple and simplicial; in H x is simplicial but not simple, since y and z are not compatible.

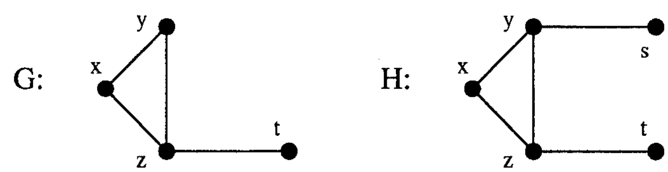

Figure 10. In G x is simple; In H x is not simple.

THEOREM II.1 (Farber) *A graph G =(V,E) is strongly chordal if and only if every induced subgraph has a simple vertex.*

For a proof of Theorem II.1 see Farber [Ref. 24]. As part of the proof of Theorem II.1, Farber introduced an algorithm that constructs a strong elimination ordering in $\mathcal{O}(n^2)$ time.

C. INTERVAL AND UNIT INTERVAL GRAPHS

Jamison and Lasker [Ref. 25] describe elimination orderings that characterize interval and unit interval graphs. Both elimination orderings can be constructed in $\mathcal{O}(n+m)$ time using algorithms due to Gavril [Ref. 26] and Booth and Leuker [Ref. 27]. For any graph G with an ordering of V as $v_1, v_2, \ldots, v_n$, the *upper* and *lower* neighborhoods $N^+[v_i]$ and $N^-[v_i]$, respectively, are defined as follows (see Figure 11):

$$N^+[v_i] = \{v_j \mid v_j = v_i \text{ or } v_i < v_j \text{ and } v_iv_j \in E\} \quad \text{(II.7)}$$

$$N^-[v_i] = \{v_j \mid v_j = v_i \text{ or } v_i > v_j \text{ and } v_iv_j \in E\} \quad \text{(II.8)}$$

The ordering $\varphi = (v_1, \ldots, v_n)$ is an *interval elimination ordering* if and only if the lower neighborhood $N^-[v_k]$ is an interval in the ordering for each v_k. That is,

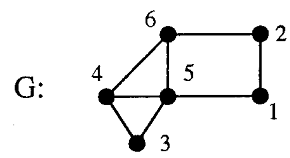

Figure 11. $N^{-}[5] = \{5,4,3,1\}$, $N^{+}[5] = \{5,6\}$.

for all v_i, v_j, and v_k with $i < j < k$, $v_iv_k \in E$ implies $v_jv_k \in E$. See Figure 12 for an example. Jamison and Lasker [Ref. 25] state the following theorem without proof; a proof can be found in Olariu [Ref. 28].

THEOREM II.2 (Olariu) *A graph $G = (V,E)$ has an interval elimination ordering if and only if G is an interval graph.*

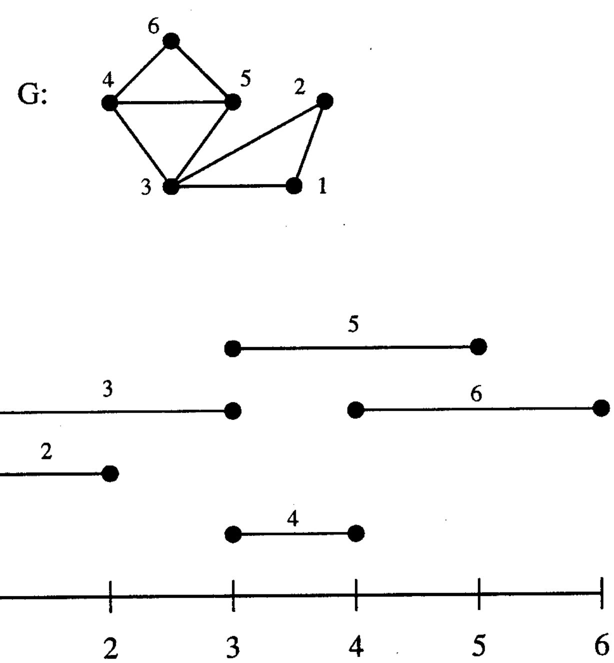

Figure 12. A graph G and an interval representation of $\{N^{-}[v_i] | v_i \in V\}$.

A *bicompatible ordering* $\rho = (v_1, \ldots, v_n)$ of a graph $G = (V,E)$ is an ordering of the vertices of V such that $(v_1, \ldots, v_n)$ and $(v_n, \ldots, v_1)$ are perfect elimination orderings for G. In other words a vertex $v_i \in$ V is simplicial in both $G - \{v_1, \ldots, v_{i-1}\}$ and $G - \{v_{i+1}, \ldots, v_n\}$. In Figure 13 both $(1,2,3,4,5)$ and $(5,4,3,2,1)$ are perfect

elimination orderings. Roberts [Ref. 29, 30] showed that a graph $G = (V, E)$ has a bicompatible ordering if and only if G is a unit interval graph.

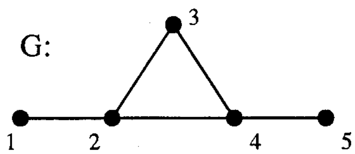

Figure 13. A unit interval graph G and a bicompatible ordering $\rho = (1, 2, 3, 4, 5)$.

D. THRESHOLD GRAPHS

A vertex v is called a *dominating vertex* if v is adjacent to every other vertex with positive degree. Therefore, if we consider the degree partition discussed in Chapter I, D_m is the set containing all of the dominating vertices of G. A *threshold elimination ordering* $\tau = (v_1, \ldots, v_n)$ is an ordering of the vertices of V such that v_j dominates all vertices of positive degree in $G - \{v_i | i > j\}$. In Figure 14, $\tau = (v_1, v_2, \ldots, v_{10}) = (7, 2, 10, 8, 5, 1, 3, 6, 9, 4)$ is a threshold elimination ordering for G.

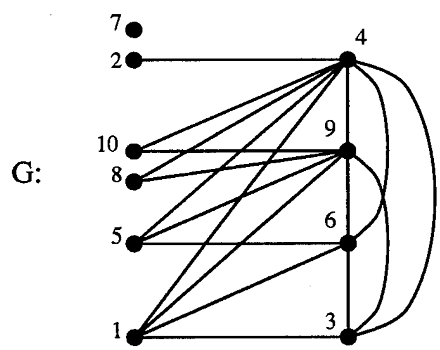

Figure 14. A threshold graph G and a threshold elimination ordering.

We will show in Theorem II.7 that if a graph G has a threshold elimination ordering then G is a threshold graph. We first prove the following lemmas that will useful to us in the proof of Theorem II.7.

Lemma II.3 *Let $G = (V,E)$ be a graph of order n, and let $\tau = (v_1, \ldots, v_n)$ be a threshold elimination ordering for G. Let $G^- = G - \{v_n\}$. Then $\tau^- = (v_1, \ldots, v_{n-1})$ is a threshold elimination ordering for G^-.*

Proof: By the definition of τ, vertex v_j dominates all vertices of positive degree in $G - \{v_i | i > j\}$. In particular, for $i = 1, \ldots, n-1$, v_j dominates all vertices of positive degree in $G - \{v_i | i > j\}$. So by letting $G^- = G - \{v_n\}$ and $\tau^- = (v_1, \ldots, v_{n-1})$, we see that τ^- is a threshold elimination ordering for G^-. □

Lemma II.4 *Let $G = (V,E)$ be a graph of order n, and let $\tau = (v_1, \ldots, v_n)$ be a threshold elimination ordering for G. Let $G^+ = G + \{v_{n+1}\}$ and suppose $\tau^+ = (v_1, \ldots, v_n, v_{n+1})$. Then $\tau^+ = (v_1, \ldots, v_n, v_{n+1})$ is a threshold elimination ordering for G^+.*

Proof: In G^+, v_{n+1} is adjacent to every vertex, and thus dominates every vertex with positive degree in G^+. Therefore, $\tau^+ = (v_1, \ldots, v_{n+1})$ is a threshold elimination ordering for G^+. □

Lemma II.5 *Let $G = (V,E)$ be a threshold graph of order n, and let $D_1, D_2, \ldots, D_m$ be a degree partition of G. Let v be a vertex in D_m. Then $G - \{v\}$ is a threshold graph.*

Proof: Suppose $v \in D_m$ and $|D_m| = 1$. Since the vertices of D_1 are adjacent only to vertices in D_m, then in $G - \{v\}$, D_1 contains only isolated vertices. Therefore, $G - \{v\}$ has degree partition $C_0, C_1, \ldots, C_{m-2}$, where $C_0 = D_0 \bigcup D_1$ and $C_i = D_{i+1}$ for $i > 0$. Let $x \in C_i$ and $y \in C_j$ in $G - \{v\}$. Now, since $xy \in E$ if and only if $i + j > m$, then $xy \in E_{G-\{v\}}$ because $(i-1)+(j-1) > m-2$. Therefore $G - \{v\}$ is a threshold graph. Now, suppose $|D_m| > 1$. Then, the vertices adjacent to v in G would have degree $\delta_i - 1$, $i = 1, 2, \ldots, m$ in $G - \{v\}$. The property, $x \in D_i$, $y \in D_j$ and $xy \in E_{G-\{v\}}$ if and only if $i+j > m$ still holds because the degree partition, $D_1, D_2, \ldots, D_m$, remains the same. Therefore, $G - \{v\}$ is a threshold graph. □

Lemma II.6 *Let $G = (V,E)$ be a threshold graph of order n, and let $D_0, D_1, \ldots, D_m$ be a degree partition of G. Join a vertex v to $G' = G - D_0$. Then the resulting graph $\{G' + \{v\}\} \bigcup D_0$ is a threshold graph.*

Proof: In the graph $G' + \{v\}$, v is adjacent to every vertex. Therefore, $v \in D_m$. Let $x \in D_i$, $x \neq v$, $i = 1, 2, \ldots, m$. Then $xv \in E_{G+\{v\}}$ and $i + m > m$ for $i = 1, 2, \ldots, m$. So, $G' + \{v\}$ is a threshold graph. Therefore, $\{G' + \{v\}\} \bigcup D_0$ is a threshold graph. $\square$

THEOREM II.7 *A graph $G = (V,E)$ has a threshold elimination ordering if and only if G is a threshold graph.*

Proof: ($\Rightarrow$) Assume G has a threshold elimination ordering. We show that G is a threshold graph, using induction on the order of V.

basis: $n = 1$. Let G be a graph with one vertex and let $\tau = (v_1)$ be a threshold elimination ordering for G. The graph on one vertex is a threshold graph so the claim holds.

inductive step: Assume that all graphs of order n that have threshold elimination orderings are threshold graphs. Let G be a graph of order $n + 1$, and let $\tau = (v_1, \ldots, v_{n+1})$ be a threshold elimination ordering for G. Then the vertex v_{n+1} dominates all vertices of positive degree in G. Let $G' = G - \{v_{n+1}\}$. By Lemma II.3, $\tau' = \tau - v_{n+1}$ is a threshold elimination ordering for G', and by the inductive hypothesis G' is a threshold graph. Therefore, by Lemma II.6, $G' + \{v_{n+1}\} = G$ is a threshold graph.

($\Leftarrow$) Assume G is a threshold graph. We show $\tau = (v_1, \ldots, v_n)$ is a threshold elimination ordering for G, using induction on the order of V.

basis: $n = 1$. Let G be the threshold graph with one vertex. Clearly, any ordering, τ, of G will be a threshold elimination ordering. So, the claim holds.

inductive step: Assume that all threshold graphs of order n have threshold elimination orderings. Let G be a threshold graph of order $n+1$. Let $D_1, D_2, \ldots, D_m$ be the degree partition for G, and let $v \in D_m$. By Lemma II.5, $G - \{v\}$ is a threshold graph, and by the inductive hypothesis $G - \{v\}$ has a threshold elimination ordering, $\tau = (v_1, \ldots, v_n)$. In G let $v = v_{n+1}$ and label the rest of the vertices with the labels given by τ in $G - \{v\}$. Since $v_{n+1} \in D_m$, then v_{n+1} is adjacent to each vertex in G. Therefore, by Lemma II.4, $\tau' = (v_1, \ldots, v_n, v_{n+1})$ is a threshold elimination ordering for G. □

In order to construct a threshold elimination ordering for a threshold graph G, we would sort the vertices of G by degree, i.e., construct the degree sequence $\mathcal{D}$. Using a bubble sort algorithm, which is easy to implement on a computer, we could construct $\mathcal{D}$ in $\mathcal{O}(n^2)$ time. If efficiency is the main concern and n is large, then the merge sort algorithm could construct $\mathcal{D}$ in $\mathcal{O}(n \log n)$ time.

Now that we have characterizations of strongly chordal, interval, unit interval, split, and threshold graphs in terms of special vertex orderings, we will use the structure of these orderings to show in Chapter III that the previously named classes of graphs are completion classes.

III. COMPLETION SEQUENCES

A. COMPLETION ALGORITHM

In this chapter we show that completion sequences can be obtained for strongly chordal, interval, unit interval, split, and threshold graphs using an Algorithms A and B. These algorithms exploit the fact that these classes of graphs can be characterized by specific vertex orderings. The use of the elimination orderings specific to strongly chordal, interval, unit interval and threshold graphs discussed in Chapter II will enable us to prove that these classes of graphs are completion classes. For split graphs we use the reverse of the degree sequence defined in Chapter I to show that the split graphs are a completion class.

ALGORITHM A
Input: Graph $G = (V, E)$ of order n and size m, with vertices labeled as $v_1, \ldots, v_n$.
Output: K_n.
BEGIN
  $G_0 =$ G;
  $E_0 =$ E;
  $s = \binom{n}{2} - m$;
  FOR i:= 1 TO s DO
    $k_i = \max\{j \mid \deg(v_j) < n - 1\}$;
    $m_i = \max\{l \mid v_{k_i} v_l \notin E_{i-1}\}$;
    $e_i = v_{k_i} v_{m_i}$;
    $E_i = E_{i-1} \bigcup \{e_i\}$;
    $G_i = (V, E_i)$;
  END FOR
END

We show in the next three sections that, given an incomplete graph $G = (V, E)$ of order n and size m that is strongly chordal, interval, or threshold, Algorithm A produces a sequence of graphs $G = G_0, \ldots, G_s$ with the same property. In each of these cases we actually prove a stronger result, that if β is an elimination ordering of the requesite type then β is an elimination ordering for each graph G_i in the sequence.

For examples see Figures 17, 18 and 19. Note that in these figures the dashed lines represent the edges added by the algorithm at each iteration.

B. STRONGLY CHORDAL COMPLETION SEQUENCES

We prove in Theorem III.2 that given a strongly chordal graph $G = (V, E)$ and a strong elimination ordering θ, constructed using the algorithm presented in Farber [Ref. 24], θ is a strong elimination ordering for each G_i produced by Algorithm A. Corollary III.3 follows immediately from the fact that G is strongly chordal if and only if G has a strong elimination ordering. We therefore conclude that the strongly chordal graphs are a completion class. We first prove the following lemma, which shows the relationship between the linear ordering of the neighborhoods of a simple vertex and the elimination ordering θ.

Lemma III.1 *Let θ be a strong elimination ordering for $G = G_0$ constructed by Farber's algorithm, and suppose that θ is also a strong elimination ordering for each of $G_1, \ldots, G_s$ constructed by Algorithm A with the specified parameters. Suppose v_x is simple in $G_i - \{v_1, v_2, \ldots, v_{x-1}\}$, where $0 \leq i \leq s$. If $v_y, v_z \in N_i(v_x)$ and $x < y < z$, then $N_i[v_y] \subseteq N_i[v_z]$.*

Proof: If $i = 0$, the result follows from a sequence of partial order relations on V constructed by Farber's algorithm. Suppose that the claim is true for some $0 \leq m < i$, and consider a simple vertex v_x that is simple in $G_{m+1} - \{v_1, v_2, \ldots, v_{x-1}\}$. Suppose $v_y, v_z \in N_{m+1}(v_x)$, with $x < y < z$. If e_{m+1} is incident to none of v_x, v_y, or v_z, or if $e_{m+1} = v_z v_w$ for some w, then the result follows from the induction hypothesis. If $e_{m+1} = v_y v_w$ for some w, then either $w = z$, which is impossible since v_x is simplicial in $G_m - \{v_1, v_2, \ldots, v_{x-1}\}$, or $w \neq z$. If $z > w$, then $N_m[z] = \mathrm{V} - \{v_1, \ldots, v_{x-1}\}$, which ensures that $N_{m+1}[v_y] \subseteq N_{m+1}[v_z]$. If $z < w$, then since $z > y$ we know that $v_z v_w \in E_m$. By the induction hypothesis $N_m[v_y] \subseteq N_m[v_z]$, and since $v_w \in N_m(z)$ it follows that $N_{m+1}[v_y] \subseteq N_{m+1}[v_z]$. The result follows by induction. $\square$

THEOREM III.2 *Let $G = (V,E)$ be a connected graph of order n and size m. Let $G_0 = G$, and define the sequence of graphs $G_0, G_1, \ldots, G_s$ using Algorithm A. If θ is a strong elimination ordering for G, then θ is a strong elimination ordering for each G_i, $i = 1, 2, \ldots, s$.*

Proof: We show that given a strong elimination ordering if v_t is a simple vertex in $G_{i-1} - \{v_1, v_2, \ldots, v_{t-1}\}$, then it is simple in $G_i - \{v_1, v_2, \ldots, v_{t-1}\}$. Note that the status of v_t is only affected if e_i is incident to a vertex in $N_i[v_t]$. Therefore, we have two cases to consider:

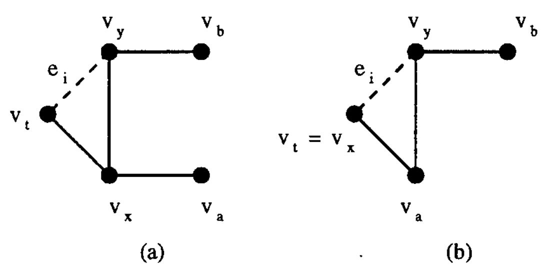

Figure 15. (a) $N_i[v_t] \neq N_{i-1}[v_t]$, (b) $N_i[v_t] \neq N_{i-1}[v_t]$, and $v_t = v_x$.

Case 1: $N_i[v_t] \neq N_{i-1}[v_t]$. Assume v_t is simple in $G_{i-1} - \{v_1, v_2, \ldots, v_{t-1}\}$ but not simple in $G_i - \{v_1, v_2, \ldots, v_{t-1}\}$. Since e_i connects v_t to some vertex v_y and since v_t has the smallest label among the remaining vertices, then $t = m_i$ and $y = k_i$. Since v_t is not simple in $G_i - \{v_1, v_2, \ldots, v_{t-1}\}$ then there exists $v_x \in N_i[v_t]$ such that v_x and v_y are incompatible. So there exists v_a and v_b such that $v_a \in N_i[v_x]$, $v_a \notin N_i[v_y]$, $v_b \in N_i[v_y]$ and $v_b \notin N_i[v_x]$. We know from our discussion of strongly chordal graphs in Chapter II that strong elimination orderings are also perfect elimination orderings. Now if $v_x \neq v_t$ then, since v_t is simplicial in $G_i - \{v_1, v_2, \ldots, v_{t-1}\}$, there exists an edge from v_y to v_x (see Figure 15(a)). Since $m_i = \max\{l | v_l v_{k_i} \notin E_i\}$, then $m_i > a$. This means that $t > a$. This contradicts the fact that $t < y$ for all $y \in \{t+1, \ldots, n\}$. Therefore, $v_t = v_x$.

So, since $v_x = v_t$ and since v_t is simplicial in $G_i - \{v_1, v_2, \ldots, v_{t-1}\}$, then the edge $v_a v_y$ is forced (see Figure 15(b)). This contradicts the assumption that v_x and v_y are incompatible. So, v_t is simple in $G_i - \{v_1, v_2, \ldots, v_{t-1}\}$.

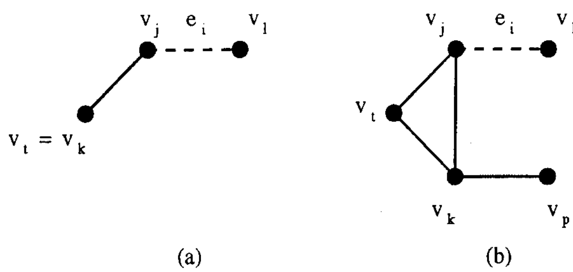

Figure 16. (a) $N_i[v_t] = N_{i-1}[v_t]$, and $v_t = v_x$, (b) $N_i[v_t] = N_{i-1}[v_t]$.

Case 2: $N_i[v_t] = N_{i-1}[v_t]$. Again assume v_t is simple in $G_{i-1} - \{v_1, v_2, \ldots, v_{t-1}\}$ but not simple in $G_i - \{v_1, v_2, \ldots, v_{t-1}\}$. As noted above, if e_i is not incident to any vertex in $N_i[v_t]$ then we are done. So we conclude that e_i is incident to some $v_j \in N_i[v_t]$. By the assumption, we know that v_t is not simple in $G_i - \{v_1, v_2, \ldots, v_{t-1}\}$, so there exists at least one $v_k \in N_{i-1}[v_t]$ such that v_j and v_k were compatible in $G_{i-1} - \{v_1, v_2, \ldots, v_{t-1}\}$, but are not compatible in $G_i - \{v_1, v_2, \ldots, v_{t-1}\}$. If $N_{i-1}[v_k] \subseteq N_{i-1}[v_j]$, then it follows that $N_i[v_k] \subseteq N_i[v_j]$, a contradiction. Therefore, we conclude that $N_{i-1}[v_k] \not\subseteq N_{i-1}[v_j]$. Let v_l be the vertex adjacent to v_j and incident to e_i in $G_i - \{v_1, v_2, \ldots, v_{t-1}\}$. Now, v_k is not adjacent to v_l otherwise we would have $N_i[v_j] \subseteq N_i[v_k]$ again contradicting the the incompatibility of v_k and v_j.

Suppose $v_t = v_k$ (see Figure 16(a)). Since v_t is simple in $G_{i-1} - \{v_1, v_2, \ldots, v_{t-1}\}$, then it is also simplicial in $G_{i-1} - \{v_1, v_2, \ldots, v_{t-1}\}$. So, $N_{i-1}[v_t]$ is a clique. Moreover, any $v \in N_{i-1}[v_t]$ is adjacent to v_j since $v_j \in N_{i-1}[v_t]$. Therefore, $N_{i-1}[v_t] \subseteq N_{i-1}[v_j]$ or $N_{i-1}[v_k] \subseteq N_{i-1}[v_j]$, a contradiction. So, we conclude that $v_t \neq v_k$ (see Figure 16(b)). If $v_t \neq v_k$, then suppose $k < j$. Since $t < k < j$, then by Lemma III.1, $N_{i-1}[v_k] \subseteq N_{i-1}[v_j]$, a contradiction. Hence $j < k$. There are two cases to consider:

Subcase A: $j = k_i$. Since $j = k_i$, then v_j is the vertex in $G_i - \{v_1, v_2, \ldots, v_{t-1}\}$ with the largest label that is not adjacent to every other vertex. Therefore, since v_k is not adjacent to v_l, then $j > k$.

Subcase B: $j = m_i$. So $l = k_i$ and v_j is the vertex with the largest label that is not adjacent to to v_l. Again, since v_k is not adjacent to v_l then $j > k$.

Therefore, we conclude that v_t is simple in $G_i - \{v_1, v_2, \ldots, v_{t-1}\}$, and this completes the proof. $\square$

Corollary III.3 *If θ is a strong elimination ordering for $G = G_0$, then G_i is strongly chordal for $i = 1, 2, \ldots, s$.*

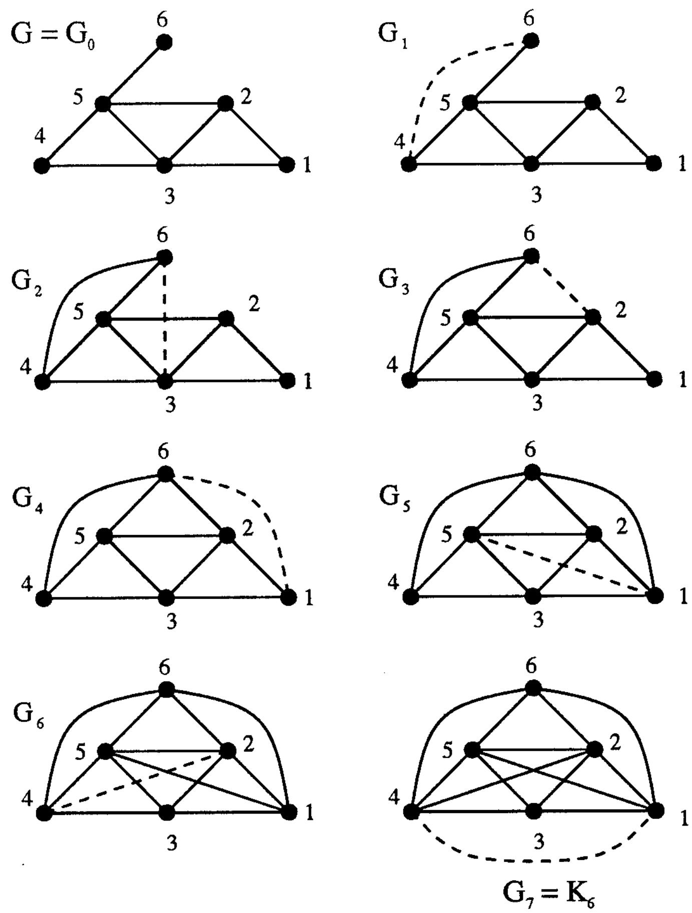

Figure 17. A strongly chordal completion sequence for G, where $\theta = (1, 2, 3, 4, 5, 6)$ is a strong elimination ordering for G. The dashed lines represent the edges added by the algorithm.

C. INTERVAL COMPLETION SEQUENCES

Using the interval elimination ordering discussed in Chapter II we show that the class of interval graphs is a completion class. See Figure 18 for an illustration of an interval completion sequence.

THEOREM III.4 *Let $G = (V,E)$ be a connected graph of order n and size m. Let $G_0 = G$ and define the sequence of graphs $G_0, G_1, \ldots, G_s$ using Algorithm A. If φ is an interval elimination ordering for G, then φ is an interval elimination ordering for each G_i, $i = 1, 2, \ldots, s$.*

Proof: Assume that φ is an interval elimination ordering for G_{i-1}. Let $v_i v_k$ be the edge chosen by Algorithm A on the i^{th} iteration of the FOR loop, with $i < k$. Assume that $N_i^-[v_k]$ in G_i is not an interval in the ordering. Then there is a vertex v_j, with $i < j < k$, such that $v_j v_k \notin E_i$. Since $i < j$, then v_i is not the vertex with the largest label such that $v_i v_k \notin E_{i-1}$ and we have a contradiction. □

Corollary III.5 *If φ is an interval elimination ordering for $G = G_0$, then G_i is interval for $i = 1, 2, \ldots, s$.*

D. THRESHOLD COMPLETION SEQUENCES

Threshold graphs are yet another class of graphs that is a completion class. We prove that if τ is a threshold elimination ordering for G then τ is a threshold elimination ordering for G_i, $i = 1, 2, \ldots, s$. See Figure 19 for an illustration.

THEOREM III.6 *Let $G = (V,E)$ be a connected graph of order n and size m. Let $G_0 = G$, and define the sequence of graphs $G_0, G_1, \ldots, G_s$ using Algorithm A. If τ is a threshold elimination ordering for G, then τ is a threshold elimination ordering for each G_i, $i = 1, 2, \ldots, s$.*

Proof: Assume that τ is a threshold elimination ordering for G_{i-1}, and suppose that τ is not a threshold elimination ordering for G_i. Then there must be a vertex, v_x,

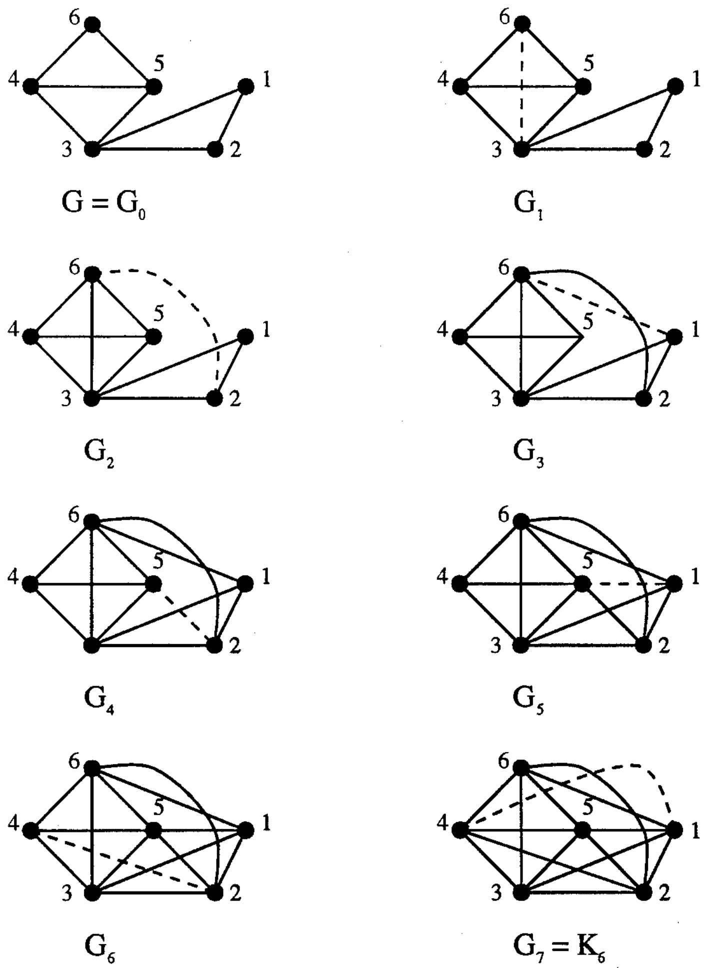

Figure 18. An interval completion sequence for G, where $\varphi = (1,2,3,4,5,6)$ is an interval elimination ordering.

that is not a dominating vertex in $G_i - \{v_{x+1}, \ldots, v_n\}$. Since v_x does not dominate in $G_i - \{v_{x+1}, \ldots, v_n\}$, then there is a vertex v_y, where $\deg(v_y) > 0$, such that $v_x v_y \notin E_i$. If $v_x v_y \notin E_i$, then $v_x v_y \notin E_{i-1}$. This contradicts the assumption that β is a threshold elimination ordering for G_{i-1}, thus completing the proof. □

Corollary III.7 *If τ is a threshold elimination ordering for $G = G_0$, then G_i is threshold for $i = 1, 2, \ldots, s$.*

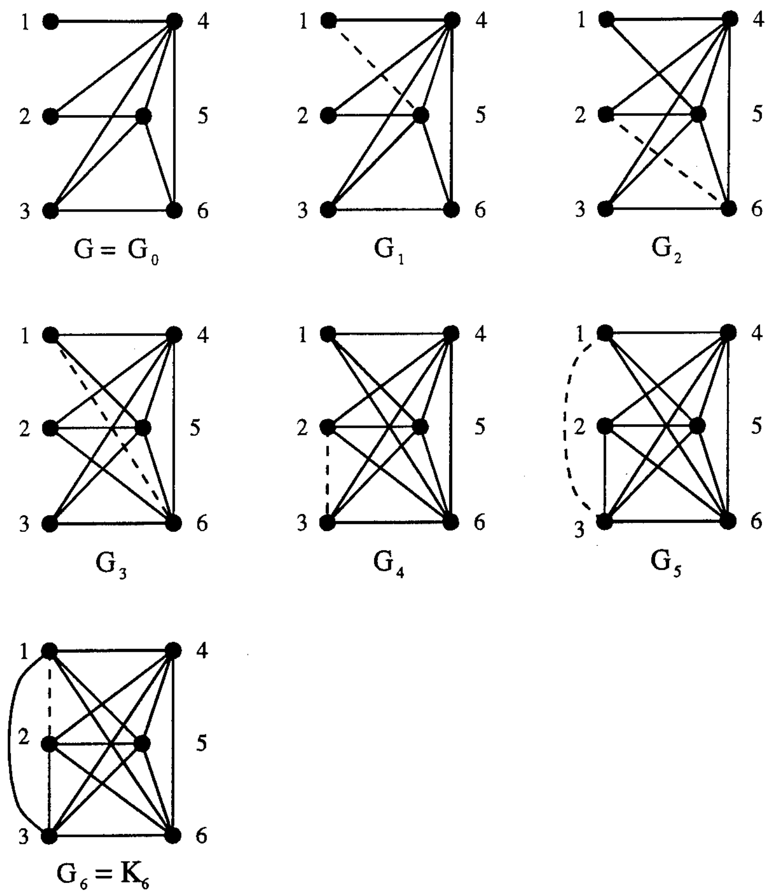

Figure 19. A threshold completion sequence for G, where $\tau = (4,5,6,3,2,1)$ is a threshold elimination ordering.

E. SPLIT COMPLETION SEQUENCES

Let $G = (V, E)$ be a graph, and let $\mathcal{D}' = (v_n, \ldots, v_1)$ be an ordering of the vertex set V such that for $i > j$, $\deg(v_i) \geq \deg(v_j)$, i.e. $\mathcal{D}'$ is the degree sequence of G in reverse order. Now we can use Algorithm A with $\mathcal{D}'$ as the ordering of the vertices of G. We show in Theorem III.8 that the sequence of graphs generated by Algorithm A are split graphs. See Figure 20 for an example.

THEOREM III.8 *Let G = (V,E) be a split graph of order n and size m. Suppose $G_0 = G$, and define the sequence of graphs $G_1, \ldots, G_s$ using Algorithm A with the ordering of the vertices of G as $\mathcal{D}'$. Then for $i = 1, 2, \ldots, s$, G_i is a split graph.*

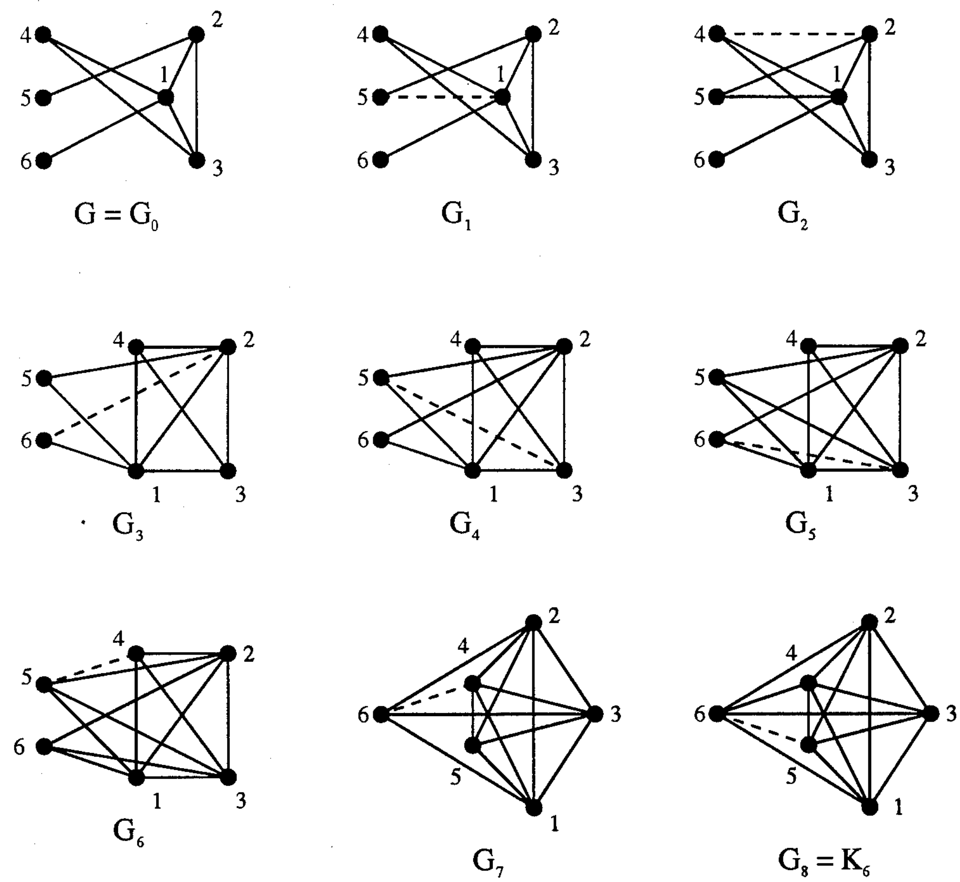

Figure 20. A split completion sequence for G, where $\mathcal{D}' = (1, 2, 3, 4, 5, 6)$ is an ordering for G by degree.

Proof: It suffices to show that at iteration i, $1 \leq i \leq s$, there is a partition of the vertex set V into a clique $K = K_{i-1}$ and an independent set $I = I_{i-1}$ such that $v_{k_i} \in K$. Let i be the smallest integer such that the graph G_i is not a split graph. Then in every partition of $V(G_{i-1})$ as $K \bigcup I$, $v_{k_i} \in I$. By our choice of k_i, since $v_{k_i} \in I$ then $\{v_{k_{i+1}}, \ldots, v_n\} \subseteq K$. There are two cases to consider.

Case 1: $K = \{v_{k_{i+1}}, \ldots, v_n\}$. In this case $v_{m_i} \in I$. Since $N_{i-1}(v_{k_i}) = K$, then the set $K' = K \bigcup \{v_{k_i}\}$ is a clique, and the set $I' = I - \{v_{k_i}\}$ is an independent set. Therefore, $V = K' \bigcup I'$ is the desired partition, contradicting our assumption.

Case 2: $\{v_{k_{i+1}}, \ldots, v_n\} \subset K$ and $v_l \in K$ for some $l < k_i$. Since $l < k_i$, then $\deg(v_l) \leq \deg(v_{k_i})$. Also $N_{i-1}(v_{k_i}) \subseteq K$ and $v_l \in K$, which implies that $\deg(v_l) \geq \deg(v_{k_i})$. Since therefore $\deg(v_l) = \deg(v_{k_i})$, then $N_{i-1}[v_l] \bigcap I = \emptyset$ so we can partition V as $K' = (K - \{v_l\}) \bigcup \{v_{k_i}\}$ and $I' = (I - \{v_{k_i}\}) \bigcup \{v_l\}$. We again obtain a contradiction to our assumption, thus completing the proof. □

F. UNIT INTERVAL COMPLETION SEQUENCES

Suppose we try to use Algorithm A to find a unit interval completion sequence for the graph in Figure 13 with the given bicompatible ordering of the vertices. Algorithm A would first choose edge 35, and we see that the resulting graph is unit interval. The next edge to be added is edge 25. Again, we see that the resulting graph is unit interval. The algorithm now adds edge 15 to the graph, and we notice that vertex 5 is no longer simplicial in the perfect elimination ordering $(5, 4, 3, 2, 1)$ because the edge 13 has not been added. So Algorithm A does not work for all unit interval graphs. Given any unit interval graph with a bicompatible ordering the next algorithm, which is a modification of Algorithm A, produces a sequence of graphs which are also unit interval.

ALGORITHM B

Input: Graph $G = (V, E)$ of order n and size m, with vertices labeled as $v_1, \ldots, v_n$.
Output: K_n.
BEGIN
    $G_0 = \mathrm{G}$;
    $E_0 = \mathrm{E}$;
    $s = \binom{n}{2} - m$;
    FOR i:= 1 TO s DO
        $k_i = \min\{k | v_k$ is not adjacent to some v_l where $l \in \{1, 2, \ldots, k-1\}\}$;
        $m_i = \max\{m | m < k_i$ and $v_{k_i} v_m \notin E_{i-1}\}$;
        $e_i = v_{k_i} v_{m_i}$;
        $E_i = E_{i-1} \bigcup \{e_i\}$;
        $G_i = (\mathrm{V}, E_i)$;
    END FOR
END

In the next theorem we show that given an incomplete unit interval graph with a bicompatible ordering $\rho = (v_1, \ldots, v_n)$, ρ is a bicompatible ordering for the sequence of graphs generated by Algorithm B. See Figure 21 for an example.

THEOREM III.9 *Let $G = (V,E)$ be a unit interval graph of order n and size m. Let $G_0 = G$ and define the sequence of graphs $G_0, G_1, \ldots, G_s$ using Algorithm B. If ρ is a bicompatible ordering for G, then ρ is a bicompatible ordering for each G_i, $i = 1, 2, \ldots, s$.*

Proof: Assume that ρ is a bicompatible ordering for G_{i-1}. Let $v_x v_y$, $x < y$, be the edge added by Algorithm B. We will show that G_i is unit interval. By way of contradiction assume that G_i is no longer unit interval. There are two cases to consider:

Case 1: v_y is not simplicial in $G_i - \{v_{y+1}, \ldots, v_n\}$. Then there exist a vertex v_k such that $v_k \in N_i^-[v_y]$ but $v_k v_x$ is not an edge in $G_i - \{v_{y+1}, \ldots, v_n\}$. By the choice of v_y, $v_1, \ldots, v_{y-1}$ is a clique. Since $x, k \in \{1, \ldots, y-1\}$, then $v_k v_x$ is an edge in $G_i - \{v_{y+1}, \ldots, v_n\}$. Therefore, v_y is simplicial in $G_i - \{v_{y+1}, \ldots, v_n\}$, a contradiction.

Case 2: v_x is not simplicial in $G_i - \{v_1, \ldots, v_{x-1}\}$. Then there exists a vertex $v_l \in N_i^+[v_x]$, such that $v_y v_l$ is not an edge in $G_i - \{v_1, \ldots, v_{x-1}\}$. If $x < l < y$ then we contradict the algorithm's choice of v_x, so $x < y < l$. Jamison and Lasker [Ref. 25] showed that if ρ is a bicompatible ordering then ρ is also an interval elimination ordering. Now, in the graph $G_{i-1} - \{v_1, \ldots, v_{x-1}\}$, $v_l \in N_{i-1}^+[v_x]$ and $v_l \notin N_{i-1}^+[v_y]$ implies ρ cannot be an interval elimination ordering, which in turn implies that ρ cannot be a bicompatible ordering. Again, we obtain a contradiction, thus completing the proof. □

Corollary III.10 *If β is a bicompatible ordering for $G = G_0$, then G_i is unit interval for $i = 1, 2, \ldots, s$.*

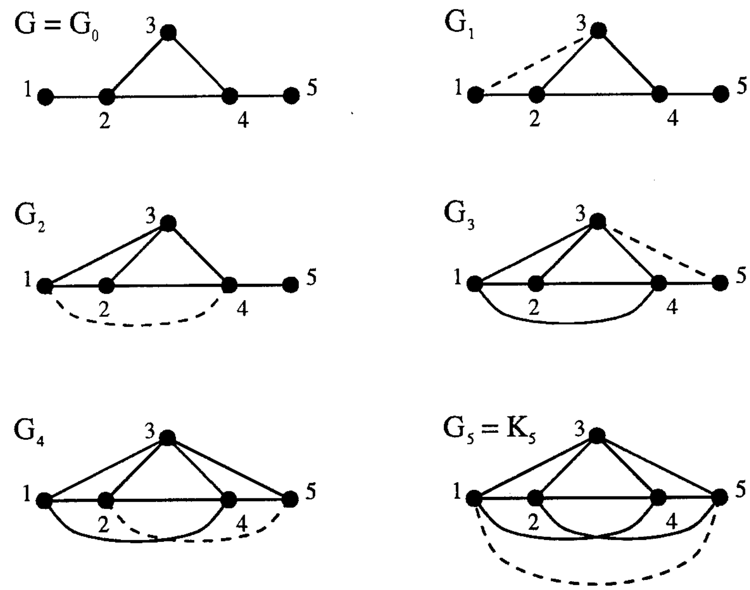

Figure 21. A unit interval completion sequence for G, where $\rho = (1, 2, 3, 4, 5)$ is a bicompatible ordering.

G. IMPLEMENTING COMPLETION ALGORITHMS

Given the appropriate ordering of the vertices of a graph, Algorithm A and Algorithm B can be implemented by the simple nested loops described in Procedure A and Procedure B respectively. In the strongly chordal, interval, split, and threshold cases we use Procedure A, while in the unit interval case we use Procedure B. The difference between A and B lies in the control loops. The complexity of each procedure is $\mathcal{O}(n^2)$.

For the strongly chordal case, we show in the next theorem that the edges added by Procedure A are precisely the same edges added by Algorithm A.

THEOREM III.11 *Let $G = (V,E)$ be a strongly chordal graph of order n and size m, and let θ be a strong elimination ordering for G. Suppose $a_1, \ldots, a_s$ and $b_1, \ldots, b_s$ are the edges added to G by Algorithm A and Procedure A respectively. Then $a_i = b_i$ for $1 \leq i \leq s$.*

Proof: Let $a_i = v_{k_i}v_{m_i}$ and $b_i = v_{x_i}v_{y_i}$ be the edges chosen at iteration i of Algorithm A and Procedure A respectively. If $a_i = b_i$, then $v_{k_i} = v_{x_i}$ and $v_{m_i} = v_{y_i}$. Let

```
PROCEDURE A(V, E, β)
BEGIN
   FOR x := n DOWNTO 2 DO
      FOR y := x − 1 DOWNTO 1 DO
         IF v_x v_y ∉ E
         THEN E := E ∪ v_x v_y;
      END FOR
   END FOR
END
```

Figure 22. Procedure A.

```
PROCEDURE B(V, E, β)
BEGIN
   FOR x := 2 TO n DO
      FOR y := x − 1 DOWNTO 1 DO
         IF v_x v_y ∉ E
         THEN E := E ∪ v_x v_y;
      END FOR
   END FOR
END
```

Figure 23. Procedure B.

$i = \min\{k | a_k \neq b_k\}$. So either $v_{k_i} \neq v_{x_i}$ or $v_{m_i} \neq v_{y_i}$. We know that $x_i \leq k_i$, since $x > k_i$ implies that $\deg_{i-1}(v_x) = n - 1$. Since $\deg_{i-1}(v_{x_i}) < n - 1$, then there exists $y < x_i$ such that $v_{x_i} v_y \notin E_{i-1}$. The absence of $v_{x_i} v_y$ would be detected by the inner loop of Procedure A, so $x_i \geq k_i$. Thus $x_i = k_i$.

If $y > m_i$, then by Algorithm A $v_{k_i} v_y = v_{x_i} v_y \in E$, so $y \leq m_i$. Since $v_{k_i} v_{m_i}$ is the missing edge, then when $y = m_i$ the absence of $v_{k_i} v_y$ will be detected by Procedure A, and so $y_i \geq m_i$. Hence, we conclude that $y_i = m_i$. Therefore, $a_i = b_i$, thus completing the proof. □

The overall complexity of constructing completion sequences for the strongly chordal, interval, unit interval, split, and threshold graphs depends on the complexity of constructing the appropriate vertex ordering. We have seen in Chapter II that

the construction of a strong elimination ordering can be done in $\mathcal{O}(n^2)$ time; the construction of interval elimination and bicompatible orderings can both be done in $\mathcal{O}(n+m)$. In the split and threshold cases, we have seen that the vertex orderings for both cases depends on sorting the vertex sets by degree; this can be done efficiently in $\mathcal{O}(n \log n)$ time. Therefore, all of these completion sequences can be constructed in $\mathcal{O}(n^2)$ time.

IV. DIRECTIONS FOR FURTHER RESEARCH

We have shown that completion sequences can be generated for the classes of strongly chordal, interval, unit interval, split, and threshold graphs by polynomial time algorithms. A new characterization of threshold graphs in terms of a threshold elimination ordering has also been presented. Listed below are some of the directions that we feel this work will lead.

• Are there other classes of graphs that can be characterized in terms of elimination orderings specific to the class? If so, can the algorithms discussed in Chapter III or a modification be used in order to to form new completion sequences? If the algorithms can not be used, do efficient algorithms exist to produce completion sequences?

• It has been shown that the chordal completion sequences are related to the problem of determining if a partial Hermitian matrix can be completed to a positive definite matrix. Is there an equivalent matrix completion problem for the strongly chordal, interval, unit interval, split, and threshold completion sequences?

• Strongly chordal graphs are associated with a particular family of integral polyhedra, and with a related family of 0,1-matrices called *totally balanced.* What significance do strongly chordal completion sequences have in these contexts?

• Can the algorithms presented in Chapter III be altered in order to produce an *annihilation sequence* of a graph $G = (V, E)$? An annihilation sequence would be a sequence of edge deletions from G such that if G has property P then each of the subgraphs in the sequence $G_1, \ldots, G_k$, where $E_k = \emptyset$, has property P. If these annihilation sequences do exist for the classes of chordal, strongly chordal, interval, unit interval, split, and threshold graphs, then given a graph G from one of these

classes, it will be possible to create a sequence $G_0, G_1, \ldots, G_i = G, \ldots, G_s$ of graphs from the class, where G_0 is the empty graph, G_s is complete, and for each $1 \leq i \leq s-1$, $G_i = G_{i-1} \bigcup e_i$ for some edge e_k. This presents the possiblity that we can view the chordal, strongly chordal, interval, unit interval, split, and threshold graphs each of order n as partially ordered under edge-set containment. We define a *partially ordered set*, or *poset* to be a pair $\mathcal{P} = (X, R)$ where the X is a set and R is a reflexive, antisymmetric, and transitive binary relation on X. Let $\mathcal{P} = (X, R)$ be a poset and suppose $x, y \in X$, $x \neq y$. We say that x and y are *comparable* in R when either xRy or yRx. We call a poset $\mathcal{P} = (X, R)$ a *chain* if every distinct pair of elements from X is comparable in R. Let $G^{(n)}$ be the set of all graphs of order n. For $G, H \in G^{(n)}$, say $G \leq H$ if and only if $E(G) \subseteq E(H)$. $(G, \leq)$ is easily seen to be a poset. If the claim is true, then each class of graphs of order n studied here would comprise a partially ordered set with the property that every graph lies on at least one chain of of length $\binom{n}{2}$. Moreover, in each of the cases in which one class properly contains another, we have containment of the corresponding posets. This phenomenon might be useful in the development of heuristics to solve general completion problems in which one seeks a graphs with some property P that is as close as possible to a particular input graph G. For more on partially ordered sets see Trotter [Ref. 31].

REFERENCES

[1] C. Berge. Farbung von Graphen, deren sämtliche bzw. deren ungerade Kreise Starr sind. *Wiss. Z. Martin-Luther-Univ., Halle-Wittenberg Math.-Natur, Reihe*, pages 114–115, 1959.

[2] L. Lovász. A characterization of perfect graphs. *Journal of Combinatorial Theory B*, 13:95–98, 1972.

[3] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, 1980.

[4] E. Marczewski. Sur deux proprietes des classes d'ensembles. *Fundamentae Mathematica*, 33:303–307, 1945.

[5] D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15:835–855, 1965.

[6] P. Hammer and B. Simeone. The splittance of a graph. Research Report CORR 77-39, University of Waterloo, 1977.

[7] A. Pnueli, A. Lempel, and S. Even. Permutation graphs and transitive graphs. *Journal of the ACM*, 19:400–410, 1972.

[8] V. Chvátal and P. Hammer. Set-packing and threshold graphs. Research Report CORR 73-21, University of Waterloo, 1973.

[9] M. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* Freeman, 1979.

[10] G. Chartrand and L. Lesniak. *Graphs and Digraphs.* Wadsworth & Brooks, 1986.

[11] F. S. Roberts. *Applied Combinatorics.* Prentice-Hall, 1984.

[12] S. Goodman and S. Hedetniemi. On the hamiltonian completion problem. In *Graphs and Combinatorics, Proc. Capitol Conf. on Graph Theory and Combinatorics, Lecture Notes in Math., vol. 406*, pages 262–272, 1974.

[13] Gavril F. Garey, M. R. and D. S. Johnson. Interval graph completion. unpublished results, 1977.

[14] F. Gavril. Path graph completion. private communication, 1977.

[15] Kaplan H. Golumbic, M. C. and R. Shamir. Graph sandwich problem. revised 1994, 1992.

[16] D. R. Lick and A. T. White. Squeeze properties for graphs. *Riv. Mat. Univ. Parma*, 11:123–131, 1970.

[17] D. R Lick and A. T. White. Chromatic-durable graphs. *Proc. Camb. Phil. Soc.*, 71:227–233, 1972.

[18] R. Grone, C. R. Johnson, E. M. Sá, and H. Wolkowicz. Positive definite completions of partial hermitian matrices. *Linear Algebra and Its Applications*, 58:109–124, 1984.

[19] C. R. Johnson. Matrix completion problems: a survey. *Proceedings of Symposia in Applied Mathematics*, 40:171–197, 1990.

[20] C. W. Rasmussen. Conditional graph completions. *Congress Numerantium*, 103:183–192, 1994.

[21] G. A. Dirac. On rigid circuit graphs. *Abh. Math. Sem. Univ. Hamburg*, 25:71–76, 1961.

[22] D. J. Rose. Triangulated graphs and the elimination process. *J. Math. Anal. Appl.*, 32:597–609, 1970.

[23] D.J. Rose, R.E. Tarjan, and G.S. Leuker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal of Computing*, 5:266–283, 1976.

[24] M. Farber. Characterizations of strongly chordal graphs. *Discrete Mathematics*, 43:173–189, 1983.

[25] R.E. Jamison and R. Laskar. Elimination orderings of chordal graphs. Technical report, Clemson University, 1983.

[26] F. Gavril. Algorithms for minimum coloring , maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM Journal on Computing*, 1:180–187, 1972.

[27] K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms. *Journal of Computer and System Sciences*, 13:335–379, 1976.

[28] S. Olariu. An optimal greedy heuristic to color interval graphs. *Information Processing Letters*, 37:21–25, 1991.

[29] F. S. Roberts. Indifference graphs. In F. Harary, editor, *Proof Techniques in Graph Theory*, pages 139–146. Academic Press, New York, 1969.

[30] F. S. Roberts. On the compatibility between a graph and a simple order. *Journal of Combinatorial Theory*, 11:28–38, 1971.

[31] W. T. Trotter. *Combinatorics and Partially Ordered Sets.* Johns Hopkins University Press, 1992.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5101	2
3.	Chairman, Code MA Department of Mathematics Naval Postgraduate School Monterey, CA 93943	1
4.	Proffessor Craig W. Rasmussen, code MA/Ra Department of Mathematics Naval Postgraduate School Monterey, CA 93943	5
5.	Proffessor David E. Mann 41 W. Elm Avenue Wollaston, MA 02170	1
6.	LT Richard M. Odom, USN Department Head Class 140 Surface Warfare Officers School Command 446 Cushing Road Newport, RI 02841-1209	2